

# Evolution of Neural Networks for Active Control of Tethered Airfoils

Allister Furey, Inman Harvey

Centre for Computational Neuroscience and Robotics  
University of Sussex, Brighton, United Kingdom  
a.d.j.furey@sussex.ac.uk, inmanh@sussex.ac.uk

**Abstract.** Recent development in tethered airfoil i.e. kite technology allows the possibility of exploitation of wind energy at higher altitudes than achievable with traditional wind turbines, with greater efficiency and reduced costs. This study describes the use of evolutionary robotics techniques to build neurocontrollers that maximize energy recoverable from wind by kite control systems in simulation. From initially randomized starting conditions, neurocontrollers rapidly develop under evolutionary pressure to fly the kite in figure eight trajectories that have previously been shown to be an optimal path for power generation. Advantages of this approach are discussed and data is presented which demonstrates the robustness of trajectory control to environmental perturbation.

## 1. Introduction

Recent advances in materials technology and kite design has facilitated the development of large scale electricity and propulsion systems that use kites to collect energy from the wind [6,7]. Here we present a simple aerodynamic simulation of a steerable four-line kite with which we use an evolutionary robotics (ER) approach in order to maximise aerodynamic forces acting along the same vector as the lines. Initially naïve neural networks are evolved using a microbial genetic algorithm [3] through selection and mutation of the controllers that produce the greatest aerodynamic forces over a given test period. Resulting controllers should steer the kite in consistent figure eight trajectories, which prior work has demonstrated are an optimal path for maximising energy recoverable from the wind [4]. The controllers should also be robust, being able to maintain stable trajectories even with significant changes in the wind velocity. As well as being a useful real world application of ER techniques, these technologies are interesting from an A-life perspective, potentially constituting self-sustaining systems through collection of the energy necessary for their operation from their environment.

The brief then for this initial study is to apply ‘off the shelf’ ER techniques to the problem of kite control in simulation, in the first instance to see if stable figure eight trajectories are evolved, and to inform further investigation.

## 2. Background

Kite energy and kite propulsion systems [5,7] currently enjoy a window of opportunity. This is largely because kite development for other applications now potentially allows the reliable production of large scale, low cost renewable energy.

The common element between current proposed traction kites applications is that the aerodynamic forces developed by the kite are transferred via the lines to perform work at near-ground level. This could be either the direct acceleration of large masses such as cargo ships [7], or the turning of a dynamo as the taut lines slowly spool out from a reel [5,6]. In this study, we focus on simply maximising the aerodynamic forces generated at the kite or maximising the component of the aerodynamic force that is in line with the flying lines. This aim is most suited to power generation applications where the direction of force is less important, as long as line tension is maintained. Marine propulsion applications will require the force in a given direction to be maximised which is to be addressed in future work.

As will be described below, the aerodynamic forces that generate line tension are proportional to the square of the apparent wind velocity at the kite. Apparent or effective wind can be defined as a vector sum of the wind speed as experienced by a stationary observer, and the negative of the movement vector of the airfoil. As the aerodynamic forces relate to the *square* of this apparent wind velocity, it is advantageous to augment the apparent wind by directing the kite to perform manoeuvres perpendicular to the wind [1]. Recent work has demonstrated that closed loop flight trajectories in the form of a figure eight are an optimal solution in terms of power generation, with the additional advantage over simple circular trajectories of not requiring a swivel mechanism to prevent crossovers accumulating in the lines [4]. If the lines are being reeled out from a spool coupled to a dynamo, the lines will need occasional retraction [4,5], at which point the line forces will need to be minimised without stalling the kite by reducing the kite speed and placing the kite perpendicular to the wind. Here, we focus purely on the power generation phase and aim to maximise forces acting through the lines.

Real world kite systems are subject to a high degree of environmental variability; wind speed, precipitation and icing will all affect the kite's performance. The real world will also present such non-linearities that current simulations are unable to render in detail without prohibitive computational costs, primarily turbulence. This issue is particularly relevant if flight is to be continued during squall or storm conditions and will become increasingly important if multiple kites are being flown in the same airspace as envisaged in recently designed kite systems [6].

Kites therefore need to be actively flown through the airspace, ideally in stereotypic trajectories, in order to maximise energy recovered from the wind. Kite power and propulsion systems also require both generation and retraction phases, upper and lower hard limits of viable operation and face a high degree of both nonlinearity and environmental variability.

Biological organisms deal with such challenges in a way that forms an effective compromise between absolute optimal solutions and solutions that are highly robust, resulting in high performance over a wide range of conditions. Additionally, evolution

can exploit aspects of the environment to improve the performance of the agent. For example, Bluefin Tuna and some flying insects actively or passively create vortices in their respective mediums to boost their performance above that anticipated by conventional fluid dynamics [12, 13]. It is this fit between the challenges of the kite control problem and the strengths of evolutionary problem solving that suggest that an ER approach is potentially fruitful. There are precedents for successful ER control of flight in both simulation [2] and in hardware [8].

## 2. Simulation Implementation

The kite itself is simulated as a curved airfoil, which viewed from the front forms a semi-circular arc (Fig.2). The kite is tethered by 4 lines and controlled from the ground by adjusting the relative lengths of the rear 2 lines. The kite is allowed to flex, without the fabric stretching and following the common Leading Edge Inflatable (LEI) kite configuration, in which the leading edge is an inflatable baton, the leading edge of the kite that normally faces into the wind is more rigid than the trailing edge opposite. Line tension and angle data is fed to the neurocontroller, which feeds back line length actuation to the kite model as per Fig.1.

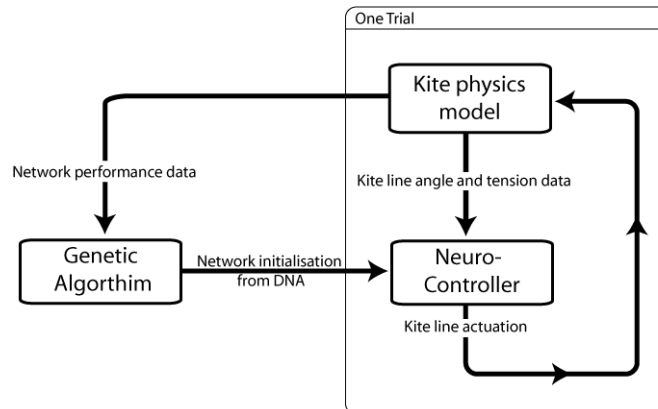
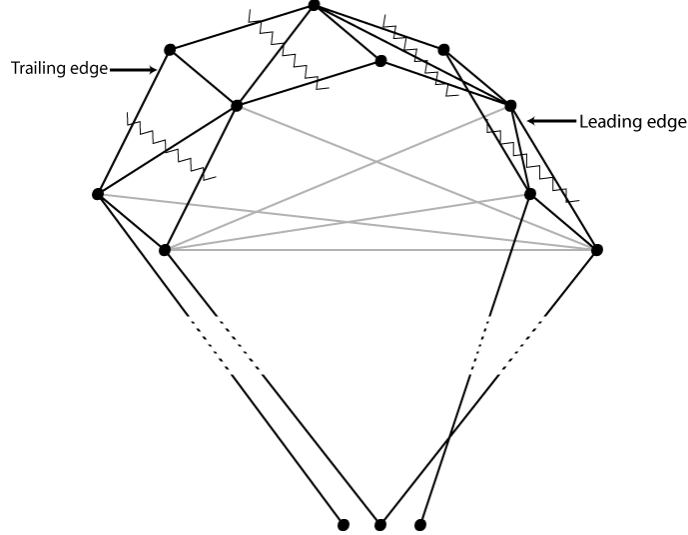


Fig. 1. A simplified schematic of the system in which neurocontrollers are evolved.

### 2.1 The Kite Physics Simulation

In contrast to other studies in which the kite is treated as single entity [1,4,5], the method of choice for this work was to use a particle based simulation. The motivation for this decision was to provide a framework in which allows explicit consideration of variation in the kite configuration in terms of kite shape, bridle setup, and physical properties of the kite such as relative rigidity and mass of kite components. Specific anticipated defects can then be introduced to the system and the adequacy of the controllers' reaction assessed. The kite is initialised as repeated rows of equidistant

particles in a semicircular arc as shown in Fig 2, which illustrates the default setup of two rows of 5 particles.



**Fig.2.** The initial configuration of particles (circles) and constraints (straight lines). The light grey constraints reinforce the arc shape of the kite and prevent ‘jellyfish’ type flapping motion, effectively performing the same role as the inflatable ribs that maintain the shape of LEI kites. The three lowest particles are the tether points. Zigzag lines indicate the positions at which the canopy is sliced for aerodynamic calculations.

For simulation purposes we approximate continuous real time by small timesteps at each of which a discrete update is made. The aerodynamic forces are calculated for each slice of the kite, as demarcated by the zigzag lines running from the leading to trailing edge in Figure 2. By slicing the kite up in this way, the force on a section of canopy depends on its particular angle of attack and the apparent wind velocity to which it is subjected. The forces are assumed to be distributable equally amongst the constituent particles, in this case pairs, of each slice although as described below this is a simplifying assumption that will be revised in future work. The acceleration on each particle is simply determined by Newton’s second law of motion (Eq.1):

$$a = \frac{f}{m} \quad (1)$$

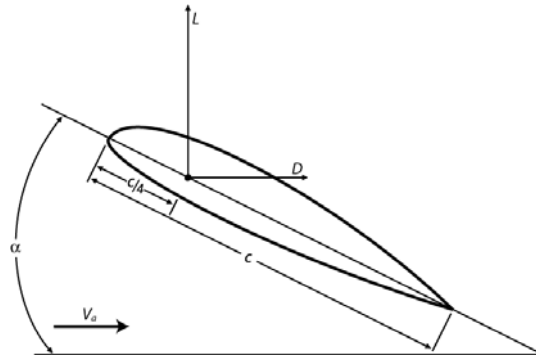
In addition to the acceleration due to aerodynamic forces, each particle is accelerated  $-9.81\text{m/s}^2$  due to gravity. Integration is performed according to the velocity free Verlet method [10] as per Eq.2; this method is used due to its relative stability and speed of execution.  $x$  here simply represents a particle position, the time step  $\Delta t$  is kept relatively small at 0.004 to avoid numerical instability.

$$x^{t+1} = 2x - x^{t-1} + a\Delta t^2 \quad (2)$$

The very diversity in forces across the canopy that is being encouraged by the slice system will quickly cause the particles to scatter, it is therefore necessary to constrain the particles to maintain the coherence of the kites structure. The constraints linking the particles are treated as infinitely stiff springs, and their positions are iterated according to the Gauss-Seidel iteration method [11]. The system in essence will simply move particles along the vector that links them in order to satisfy the constraint, i.e. two particles 2 units apart but with a constraint distance of 4 will each be moved 1 unit away from their original positions along the vector between them. By iterating around the set of constraints a number of times the system can be forced to maintain its exact initial configuration. Here the iteration number is set to one, which allows the trailing edge of the canopy that is less constrained to flex more easily than the front, which will itself flex slightly at the upper range of aerodynamic forces. The system can be set to respect the relative masses of the particles and this is the default in this implementation. It is also possible to allocate particles to represent the lines and therefore include the effects of their drag and momentum upon the kite and additionally allow sagging of the lines when under low tension. This was avoided in this initial study due to the additional computational overhead. The single constraints that constitute each line are one-way, only being enforced when the lines exceed their initial length and not when the lines are effectively slack.

### 2.1.1 Aerodynamic Model

The forces upon each slice are determined according to a simplified aerodynamics model (see Fig. 2).



**Fig.2.** A diagram of the principal aerodynamic forces upon a 2-d airfoil section. The term  $\alpha$  denotes the angle of attack, the angle at which the airfoil is inclined relative to the apparent wind,  $c$  the airfoil chord,  $V_a$  the apparent wind velocity and  $L$  and  $D$  the Lift and Drag forces respectively. Adapted from [9].

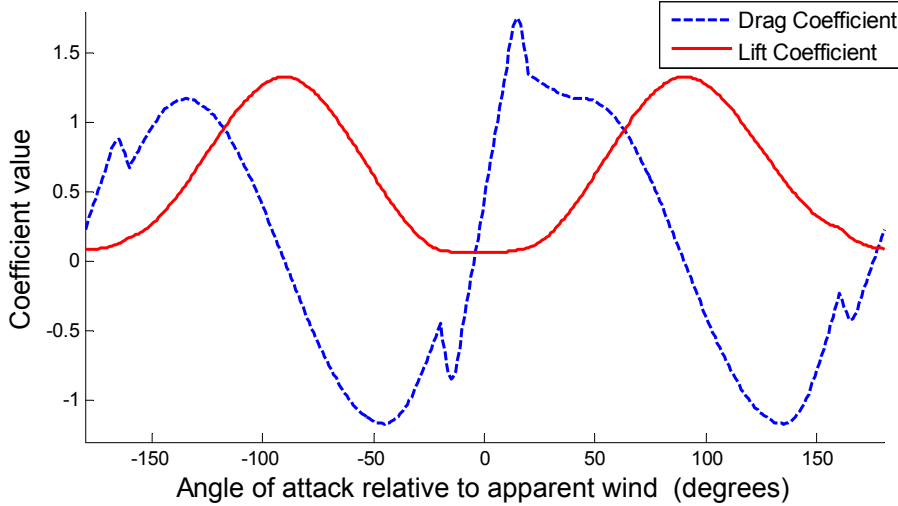
This model is simplified in that no moment coefficient is used and all forces are applied equally amongst the slice's constituent particles. The lift value for each slice is derived through Eqs.3 and 4, the lift force always acts in a direction perpendicular to the apparent wind vector. In the model, the direction of the lift force vector  $F_L$  is given by the cross product of the apparent wind vector and the vector that describes the slice's leading edge,  $a$  and  $e$  respectively in Eq.4. The drag force  $F_D$  is always in line with the apparent wind vector; therefore a unit vector of the apparent wind directs

the drag force upon the particles as per Eq.5.  $A$  is the slice's area,  $V_a$  the apparent wind velocity,  $d$  the air density,  $C_L$  and  $C_D$  are the lift and drag coefficients at the current angle of attack.

$$F_L = \bar{L} \frac{1}{2} C_L(\alpha) d V_a^2 A \quad \bar{L} = \frac{a}{\|a\|} \times e \quad (3, 4)$$

$$F_D = \frac{a}{\|a\|} \frac{1}{2} C_D(\alpha) d V_a^2 A \quad (4)$$

The lift and drag coefficients are read from a lookup table according to the slice angle of attack, values are plotted over all angles in Fig.3. Values were generated using the X-plane<sup>®</sup> Airfoil Maker version 860 software, using typical traction kite characteristics of moderate camber and thickness and relatively high drag and lift.



**Fig.3.** Graph describing drag and lift coefficient changes over change in angle of attack relative to apparent wind

In this preliminary investigation, angle of attack cannot be actively changed in order to best correspond to our prototype hardware system, although this can be achieved by changing the relative lengths of the front and back lines. A simple arbitrary gust generation model is used that generates deviations around a base windspeed of 8m/s. Each timestep has a small probability that a gust or lull is initiated, if so its maximum deviation, onset and decay speeds are set. At each subsequent timestep the windspeed is altered by a small portion of the difference between the current windspeed and the predetermined maximum gust/lull value. No lateral deviation to the wind vector is implemented, although this and the use of real recorded wind records would be logical extensions.

## 2.2 Neurocontrollers

Two classes of neurocontroller were assessed, both are small recurrent neural networks of 5 input neurons and 7 fully connected interneurons, both inhibitory and excitatory connections are permissible. In these experiments only data measurable with line angle and tension sensors at ground level is made available to the network, as described in Table 1. For inputs we make the simplifying assumption that line tension will be proportional to the aerodynamic forces generated at the kite.

Input No.	Input data
1	Total Line Tension
2	Tension difference between left and right line sets
3	Average line azimuth
4	Average line elevation
5	Difference in elevation between left and right steering lines

**Table 1.** Input data available to the neurocontrollers, all sensory data is subject to low level gaussian noise.

The simplest neurocontroller class was a discrete time recurrent network whose nodes' activation value at a given timestep  $t$  ( $\Delta t = 0.004$ ) is given by Eq.6:

$$a_j^t = \sigma\left(\sum w_{ij} a_i^{t-1}\right) - \theta_j \quad (6)$$

The second neurocontroller class was a continuous time recurrent network (CTRNN), a single neuron derives its dynamics from Eq.7:

$$\tau_j a_j = -a_j + \sum w_{ij} \sigma(o_i - \theta_i) \quad (7)$$

In both cases  $\theta$  is the bias term,  $w$  the weight and  $\sigma$  is the sigmoid function which scales values to a range between 0 and 1. In the CTRNN equation,  $\tau_j$  is the neurons time constant, and  $o$  the old activation value from the previous timestep. CTRNN neurons are integrated using Euler integration at the same timestep as the physics simulation. One of the neurons is chosen to be the output neuron, and 0.5 subtracted from its sigmoided output. The motor position is modified by 1% of the difference between the network output and its current position each millisecond. This process has three consequences; the motor output is shielded from the majority of noise in the network, the motors are prevented from moving at speeds that are unrealistically fast and finally the extremes of motor output that correspond to a 1m difference in line lengths, usually destabilising for the kite, are only rarely achieved.

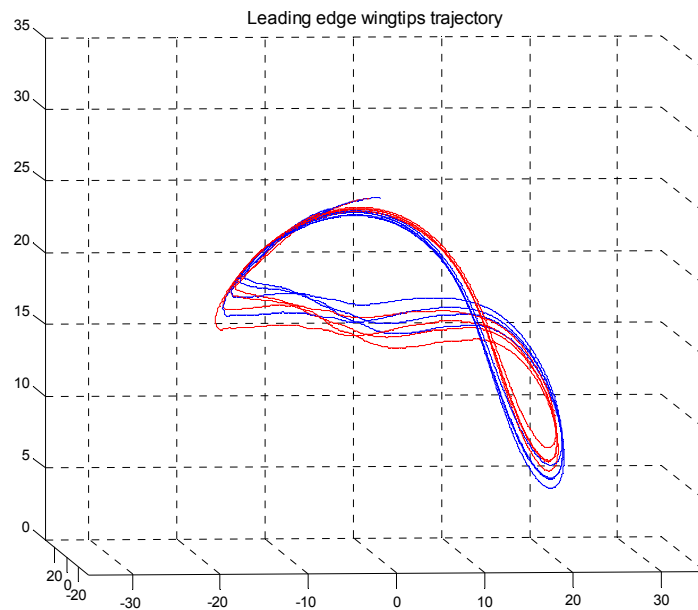
## 2.3 Genetic Algorithm

The genetic algorithm (GA) is a simple tournament based microbial GA [3], with DNA strings composed of real values determining weights both between all neurons and sensory inputs, thresholds and in the CTRNN case, time constants for each neuron in the network. The GA uses the gust generator to generate a wind trace, two

individuals are then selected at random from the population and their fitness, either the average of the aerodynamic forces produced by the whole kite, or the component of the aerodynamic force in line with the lines, is determined using the identical wind trace. The kite is always initialised at zenith position, i.e. directly above the tether points. To bias the solutions against brushes with the ground, the test is terminated prematurely as soon as any component of the kite meets ground level, regardless of whether the kite is capable of proceeding in flight or not. The individual with least fitness has its DNA string copied over by the winner with a small mutation applied to every value in the string.

### 3. Results

The key result derived from this initial investigation is that a simple 7 interneuron network is able to control a simulated kite to fly in figure 8 trajectories in only 200 generations with a population size of 20. With a trial time of 42 seconds, this corresponds to less than 47 hours of real world flight time.



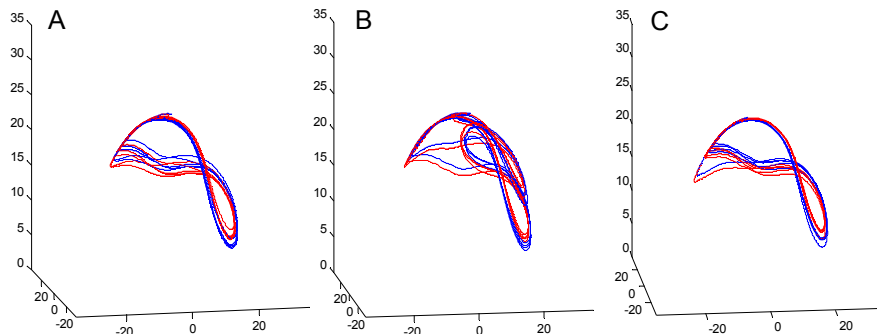
**Fig.4.** The leading edge wingtips flight trajectory over a 42 second trial period, leading edge is tethered at  $[0,0,0]$ .

The discrete-time neural networks found the best solutions in this limited evolution period (Fig.4). The reason for their outperformance of the CTRNN networks is currently unclear, but is potentially due to the additional time required for the selection of satisfactory time constants, or the fact that the CTRNNs were not initialised under centre crossing conditions [14]. The improvement in fitness across



generations, with both neuron classes, does not plateau after 200 generations suggesting that additional evolutionary time would result in further improvement in performance. Whether the fitness function was dependent on the total force generated at the airfoil or just the component of that force in line with the lines had no significant effect on the form of the trajectory, although controllers evolved under the latter condition performed significantly more figure repetitions within the trial period.

It is noteworthy that in contrast to other work [4], the evolved trajectories tend to use the whole wind window and not a small arc of less 0.3 radians directly downwind of the controller. The much shorter line length in this model (25m here vs. 100m in [4]) may have contributed to this difference. Additionally, the trajectories evolved often were, as per Fig.4, not centred directly downwind of the tether point. This may have been due to the restriction of actuation speed imposed by the motor output mapping process but is more likely to be an artefact of evolution or the test regime. Early successful controllers from an initial population are more likely to stay in flight if they swoop down to one side, where the kite naturally slows, than swoop down to the centre where the kite continually accelerates and is very sensitive to input. This effect will be addressed in further work, potentially by initialising the kite in a different position at the start of each trial.



**Fig.5.** Flight trajectories using the same controller as that in Fig. 4. A reflects 40 seconds of unchanging wind velocity, B reflects a subsequent 40 seconds of high gusts and C a subsequent return to constant wind speed.

The controller whose flight trajectories are shown in Fig.4 and 5 is robust to all but the most severe gusts and lulls from 5 to 11m/s, around a base wind speed of 8m/s. In smaller or slower wind variation, the evolved trajectory is maintained with small variations. Fig 5 illustrates the initial trajectory (A) deviating to a disturbed state during severe fast onset and offset gusts (B) and subsequently returning to its original trajectory (C) during three 40 second phases of a 120 second trial. As expected, variation in the wind speed during the evolutionary trials was important in rendering the controllers robust to gust and lulls. Only slight gusts and lulls would cause controllers evolved in constant winds to crash.

## 4. Conclusion

This study has demonstrated that the application of ER techniques to kite control produces controllers that fly the kite in stable figure eight trajectories, shown previously to be an optimal path for recovery of energy from the wind. Evolved neurocontrollers robustly maintain these trajectories during significant deviations of wind speed. These results suggest that it is worth pursuing ER for kite control systems, where real world implementations would have implications both for commercial power generation and for the capability of autonomous artificial agents to source their own power.

## 5. Acknowledgements

Thanks must go to J.R. Gludemans for his advice regarding kite simulation. I am also grateful to the KiteGen team who welcomed my contribution to their project. I must also thank Boris Houska who provided me with a pre-print version of his paper.

## 6. References

1. Loyd M, 1980 Crosswind Kite Power, *Journal of Energy*, 4(3) 106-111 AIAA
2. Shim YS, Kim CH, 2006 Evolving Physically Simulated Flying Creatures for Efficient Cruising. *Artificial Life* 12(4), MIT Press.
3. Harvey, I. 2001, Artificial Evolution: a Continuing SAGA, In *Evolutionary Robotics: From Intelligent Robots to Artificial Life*, T. Gomi (ed.) Springer-Verlag.
4. Houska B, Diehl M, 2007 Optimal Control for Power Generating Kites, 2007 *European Control Conference*, proceedings not yet in publication.
5. Canale M, Fagiano L, Ippolito M, Milanese M, 2006 Control of tethered airfoils for a new class of wind energy generator. *45th IEEE Conference on Decision and Control* proceedings not yet in publication.
6. Ippolito M, 2006 Vertical axis wind turbine with control system steering kites, Patent Publication number EP1672214
7. Wrage S, Mueller S. 2007 Watercraft comprising a free flying kite-type wind attacked element as a wind powered drive unit, Patent Publication number KR20070007342
8. J-C Zufferey, A Klaptocz, A Beyeler, JD Nicoud, and D Floreano. 2006 A 10-gram microflyer for vision based indoor navigation. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'06)*
9. Glauert H, *The Elements of Airfoil and Airscrew Theory*, 1959, Second Edition, Cambridge University Press.
10. Verlet L, 1967 Computer experiments on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules, *Phys. Rev.*, 159, 98-103
11. Press WH, 1993 *Numerical Recipes*, Cambridge University Press
12. Techet AH, Hover FS, Triantafyllou MS, Separation and Turbulence Control in Biomimetic Flows, 2003 *Flow, Turbulence and Combustion*, 71 105-118
13. Ellington CP, The Novel Aerodynamics of Insect Flight, 1999 *J. Exp Biol.*, 202 3439-3448
14. Mathayomchan B, Beer RD, 2002, Center-Crossing Recurrent Neural Networks for the Evolution of Rhythmic Behavior, *Neural Computation*, 14(9), 2043-2051